

# Using the Cluster - GPU Scheduling

At the time of this document, RC has three models of nVidia GPU in the compute cluster:

- gt740, 2Gb
- k20, 4.8Gb
- Titan, 6Gb

These GPUs are scheduled using the **Generic REsource Scheduler** (gres) feature of slurm. This ensures that the only jobs that can talk to a GPU are the ones scheduled against it. We made a design decision to put all GPUs in a single gres called "gpu" even though we have two basic sizes. If you simply request a gpu gres, you may get a small or large card depending on availability.

We leverage a second feature of slurm called *constraints* to restrict which nodes are eligible to service a job request based on features we assign to specific nodes. Some of the current features include: rhel, rhel6, rhel7, intel, amd, cuda, bigcuda, infiniband.

The two features we are interested in for GPU scheduling are *cuda* and *bigcuda*. Nodes with small GPUs (gt740) have a feature of *cuda*. Nodes with big GPUs (k20, titan) have a feature of *bigcuda*. This way, you can constrain your job to an appropriately sized GPU.

## Examples

**Run the job file SmallTraining.sh against any gpu available in the cluster**

```
sbatch --qos=work --gres=gpu SmallTraining.sh
```

**Run the job file LargeTraining.sh against large memory GPUs**

```
sbatch --qos=work --gres=gpu --constraint=bigcuda LargeTraining.sh
```

## Related articles

- [Slurm Basic Commands](#)
- [Submitting a Job with sinteractive](#)
- [Using GPUs with --gres](#)
- [MATLAB](#)
- [Using the Cluster - GPU Scheduling](#)